University of Colombo School of Computing
SCS 3213 / IS 3112 / CS 3213 - Game Development

Practical sheet 03

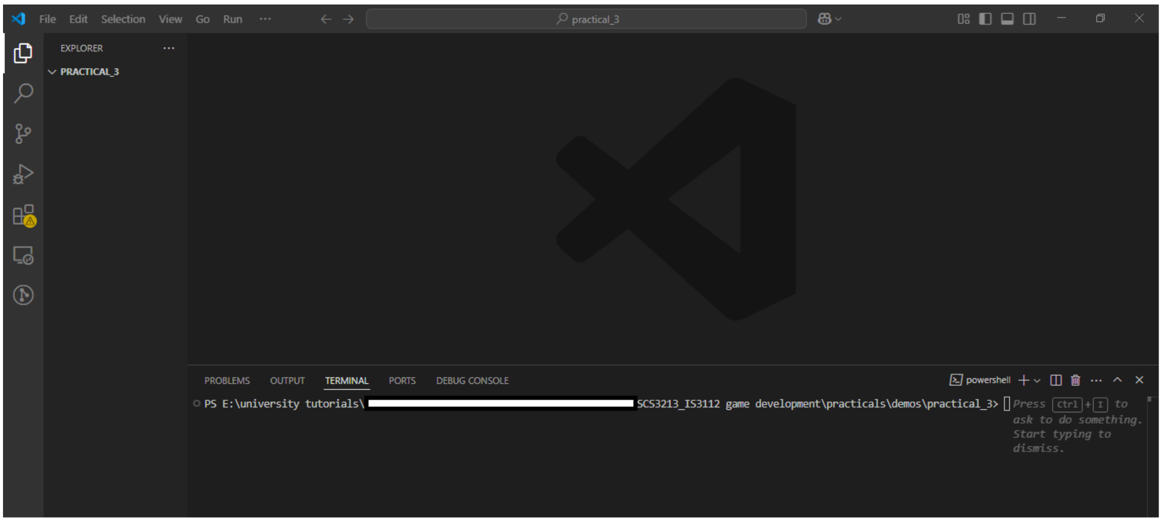
Three.js

Materials

Example:

Create a material using the “MeshDepthMaterial” material type and apply it to a cube geometry. (You can refer to the last week’s practical to get an idea about the code snippets that should be used here)

1. Create a folder, e.g., Practical_3



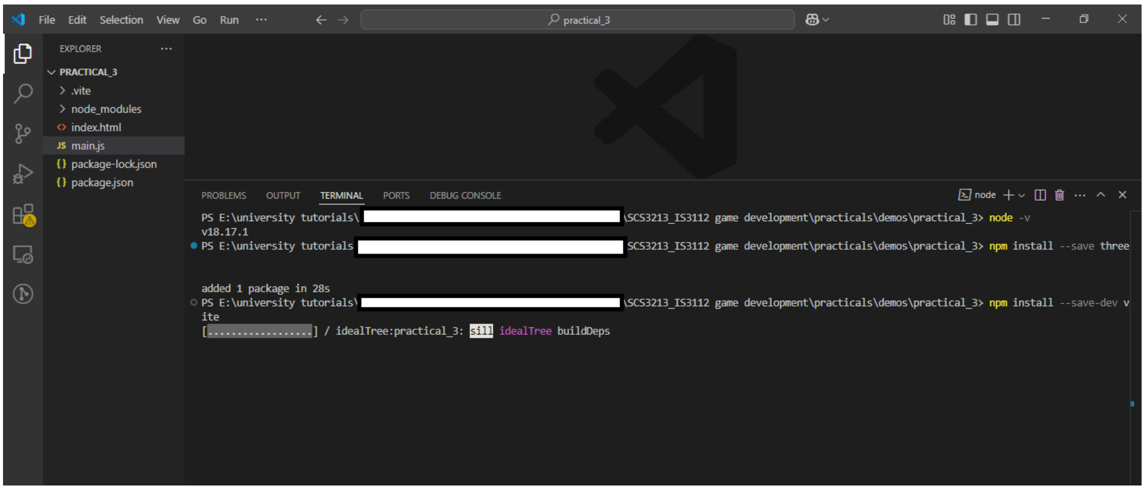
```
PS E:\university tutorials\[redacted]SCS3213_IS3112 game development\practicals\demos\practical_3> Press Ctrl+I to ask to do something. Start typing to dismiss.
```

2. Run the following codes to prepare the initial environment to run Three.js program

To check the installed node version- > *node -v*

To install Three.js to the created folder -> *npm install –save Three*

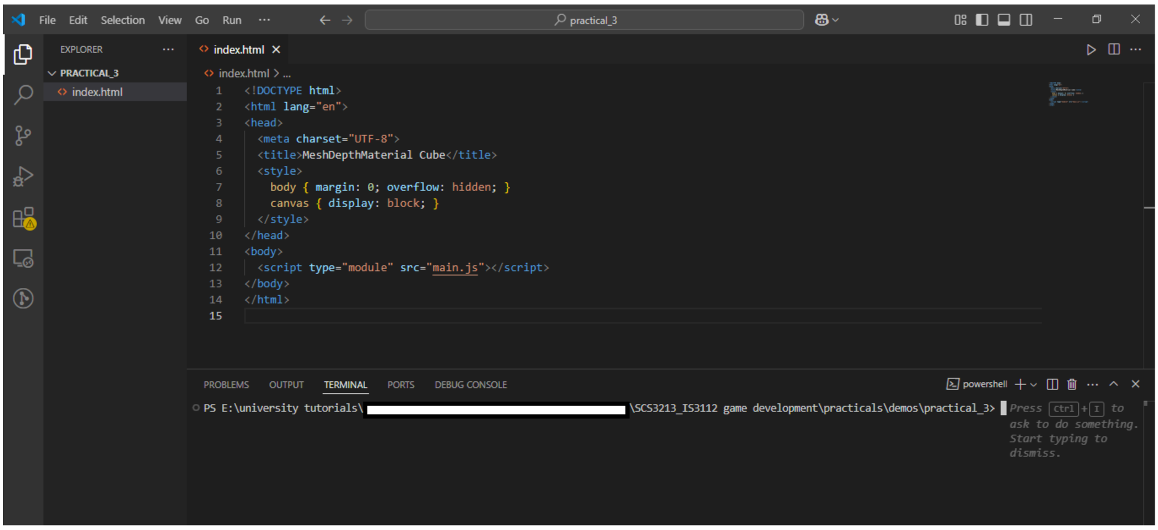
To install vite editor -> *npm install –save-dev vite*



```
PS E:\university tutorials\[redacted]\SCS3213_IS3112 game development\practicals\demos\practical_3> node -v
v18.17.1
PS E:\university tutorials [redacted]SCS3213_IS3112 game development\practicals\demos\practical_3> npm install --save three


added 1 package in 28s
PS E:\university tutorials\[redacted]\SCS3213_IS3112 game development\practicals\demos\practical_3> npm install --save-dev v
ite
[.................] / idealTree:practical_3: sill idealTree buildDeps
```

3. Inside the “Practical_3” folder, create a file: index.html. And paste the following code snippet there.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>MeshDepthMaterial Cube</title>
  <style>
    body { margin: 0; overflow: hidden; }
    canvas { display: block; }
  </style>
</head>
<body>
  <script type="module" src="main.js"></script>
</body>
</html>
```



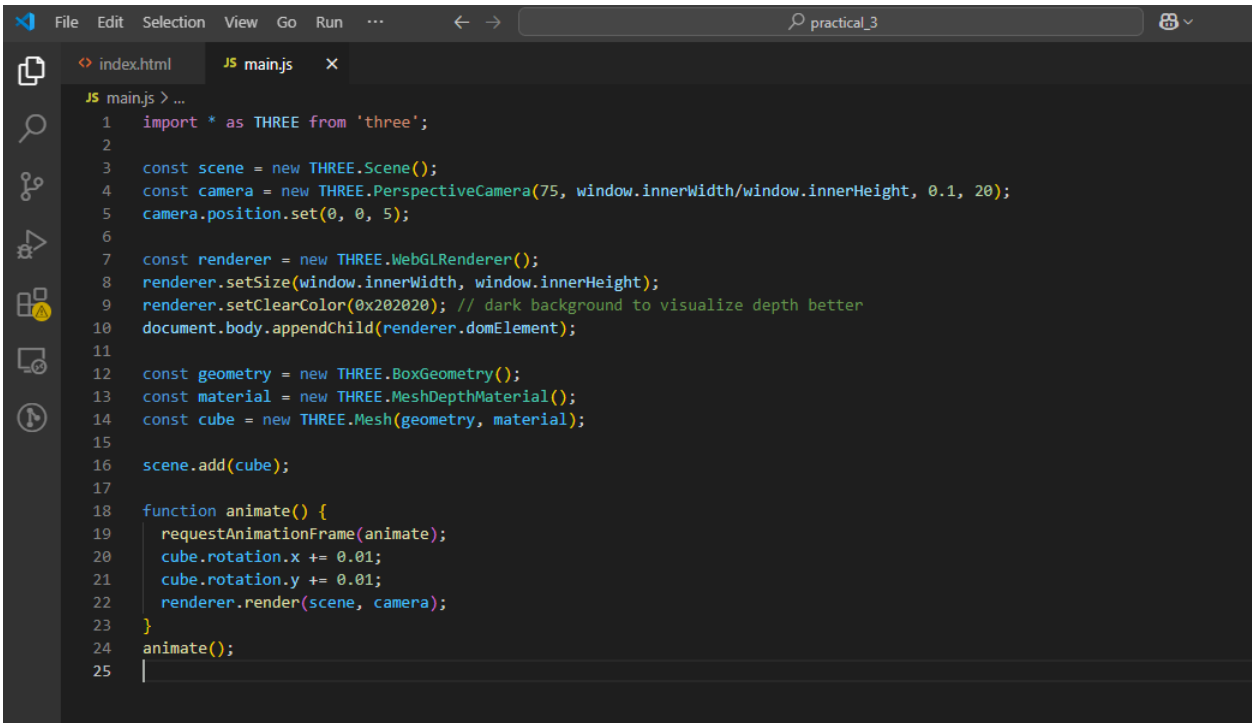
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>MeshDepthMaterial Cube</title>
  <style>
    body { margin: 0; overflow: hidden; }
    canvas { display: block; }
  </style>
</head>
<body>
  <script type="module" src="main.js"></script>
</body>
</html>
```

```
PS E:\university tutorials\[redacted]\SCS3213_IS3112 game development\practicals\demos\practical_3> Press ctrl + I to ask to do something. Start typing to dismiss.
```

4. Create another file: main.js.

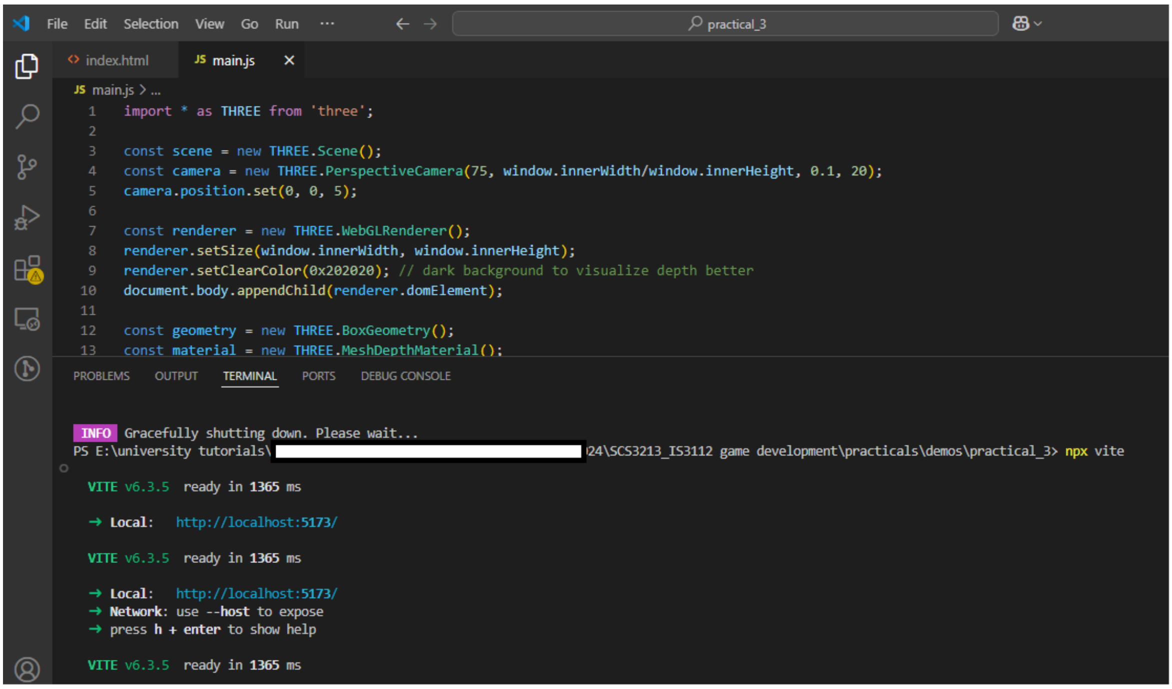
```
import * as THREE from 'three';

const scene = new THREE.Scene();
const camera = new THREE.PerspectiveCamera(75,
window.innerWidth/window.innerHeight, 0.1, 20);
camera.position.set(0, 0, 5);

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x202020); // dark background to visualize depth better
document.body.appendChild(renderer.domElement);

const geometry = new THREE.BoxGeometry();
const material = new THREE.MeshDepthMaterial();
const cube = new THREE.Mesh(geometry, material);

scene.add(cube);
```

```
function animate() {
  requestAnimationFrame(animate);
  cube.rotation.x += 0.01;
  cube.rotation.y += 0.01;
  renderer.render(scene, camera);
}
animate();
```



```
import * as THREE from 'three';

const scene = new THREE.Scene();
const camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 20);
camera.position.set(0, 0, 5);

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x202020); // dark background to visualize depth better
document.body.appendChild(renderer.domElement);

const geometry = new THREE.BoxGeometry();
const material = new THREE.MeshDepthMaterial();
const cube = new THREE.Mesh(geometry, material);

scene.add(cube);

function animate() {
  requestAnimationFrame(animate);
  cube.rotation.x += 0.01;
  cube.rotation.y += 0.01;
  renderer.render(scene, camera);
}
animate();
```

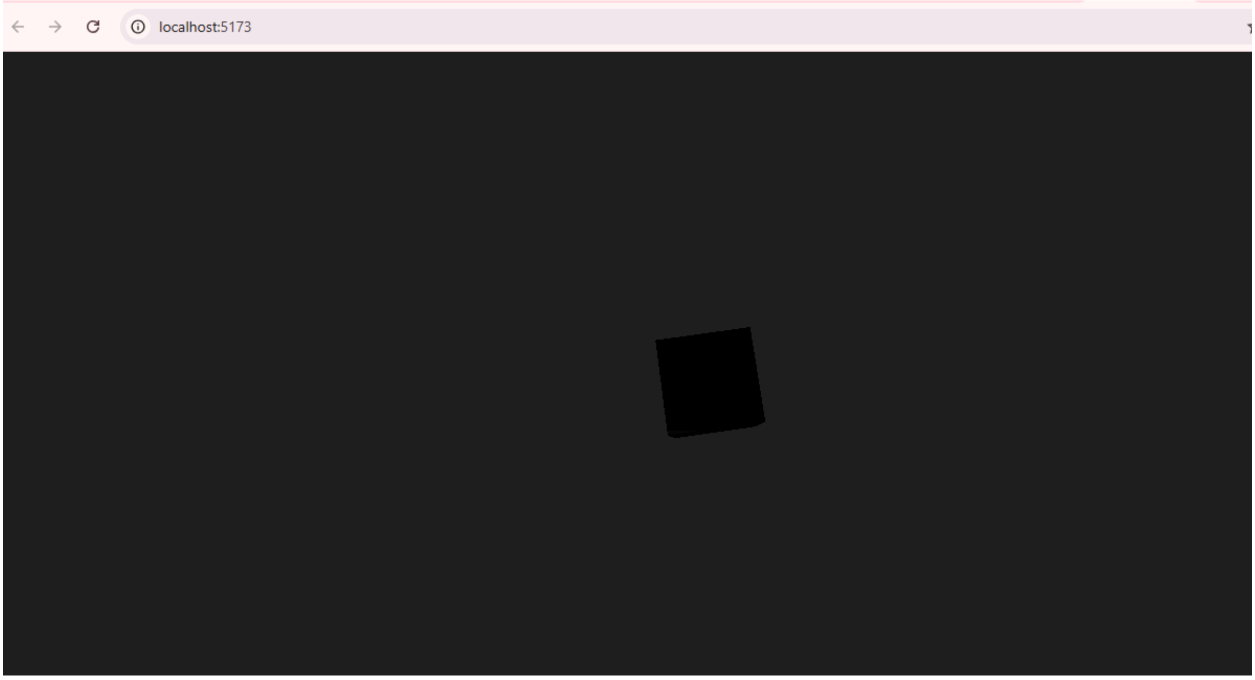
5. Run “npx vite” to run the program


File Edit Selection View Go Run ···
practical_3
index.html
main.js
main.js > ...
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
import * as THREE from 'three';

const scene = new THREE.Scene();
const camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 20);
camera.position.set(0, 0, 5);

const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x202020); // dark background to visualize depth better
document.body.appendChild(renderer.domElement);

const geometry = new THREE.BoxGeometry();
const material = new THREE.MeshDepthMaterial();
```

```
 INFO  Gracefully shutting down. Please wait...
PS E:\university tutorials\[redacted]24\SCS3213_IS3112 game development\practicals\demos\practical_3> npx vite

  VITE v6.3.5  ready in 1365 ms

  ➜  Local:   http://localhost:5173/

  VITE v6.3.5  ready in 1365 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help

  VITE v6.3.5  ready in 1365 ms
```

Output:



Activity 01

What is the difference between MeshBasicMaterial and MeshPhongMaterial in three.js?

<u>Camera Controls</u>

Activity 03

How do you initialise TrackballControls for a camera in three.js?

Activity 04

Which properties of TrackballControls can be adjusted to change its behaviour? Provide an example.

Activity 05

Implement fly controls to navigate the scene with Three.js. Use “THREE.FlyControls” to create a first-person flight experience. They should initialise the controls and update them with a clock for timing.

Activity 06

Applying a Basic Texture to a Mesh

- **Task:** Write a function to apply a basic texture to a cube mesh in Three.js.
- **Instructions:**
    1. Create a cube geometry.
    2. Load a texture image and apply it to a mesh material.
    3. Create a mesh using the geometry and material, and add it to the scene.

Activity 07

Adding a Bump Map to a Mesh

- **Task:** Implement a function to add a bump map to a cube mesh in Three.js.
- **Instructions:**
    1. Create a cube geometry.
    2. Load a texture image and a bump map image.
    3. Apply both the texture and the bump map to a mesh material.
    4. Create a mesh using the geometry and material, and add it to the scene.

Activity 08

Creating a Skybox

- **Task:** Write a function to create a skybox in Three.js.
- **Instructions:**
    1. Load six texture images for the six faces of the skybox cube.
    2. Create materials for each face using `THREE.MeshBasicMaterial`.
    3. Apply the materials to a box geometry and add it to the scene.

Activity 09

Loading and Displaying a GLTF Model

- **Task**: Load a `.gltf` model using `THREE.GLTFLoader` and add it to your scene with specific transformations.
- **Steps**:
    1. Use `THREE.GLTFLoader` to load the model.
    2. Add the model to the scene.

3. Apply transformations such as scaling and positioning to the model.

Activity 10
Adding a Normal Map to a Mesh

- **Task**: Create a mesh with a normal map to enhance the details of the surface.
- **Steps**:
    1. Load the texture and normal map using `THREE.TextureLoader`.
    2. Create `THREE.MeshPhongMaterial` and set the `normalMap` property.
    3. Create a geometry and mesh, then add it to the scene.